基于相关运算的动态目标跟踪系统开发

**学 院：计算机与大数据学院**

**专 业：人工智能**

**姓 名：陈亚欣**

**学 号：102204223**

**2025年 5 月 2 日**

基于相关运算的动态目标跟踪系统开发

摘 要

**本项目目标为在不使用AI算法进行目标识别、判别或分类的前提下，选择并编写实现合适的匹配算法，实现动态目标跟踪系统开发。系统尝试使用归一化互相关（Normalized Cross-Correlation，NCC）、差值平方和（Sum of Squared Differences，SSD）作为主要匹配算法，以卡尔曼滤波作为辅助实现目标的识别与跟踪，提升了系统在无深度学习模型的条件下目标匹配的鲁棒性与准确性。最后进行可视化与输出量化指标，绘制出目标运动轨迹并保存，输出漏检率等指标。**

**关键词： 目标识别与跟踪、NCC、SSD、卡尔曼滤波、无深度学习模型、可视化**

目 录

[第1章 方法设计 4](#_Toc197524285)

[1.1算法流程图 4](#_Toc197524286)

[1.2跟踪策略 4](#_Toc197524287)

[1.3核心算法说明 4](#_Toc197524288)

[1.3.1 NCC（归一化互相关） 4](#_Toc197524289)

[1.3.2 SSD（差值平方和） 5](#_Toc197524290)

[1.3.3 卡尔曼滤波 5](#_Toc197524291)

[1.3.4 多模板匹配 5](#_Toc197524292)

[1.3.5 局部搜索 6](#_Toc197524293)

[1.3.6 模糊搜索 6](#_Toc197524294)

[第2章 系统实现 6](#_Toc197524295)

[2.1开发环境 6](#_Toc197524296)

[2.2代码结构与关键函数示例 7](#_Toc197524297)

[第3章 实验与结果 7](#_Toc197524298)

[3.1测试案例对比 7](#_Toc197524299)

[3.1.1 NCC与SSD 8](#_Toc197524300)

[3.1.2 不同模板选取 8](#_Toc197524301)

[3.1.3 是否使用卡尔曼滤波 9](#_Toc197524302)

[3.2量化性能分析 10](#_Toc197524303)

[3.3可视化效果 10](#_Toc197524304)

[3.3.1 视频帧处理 10](#_Toc197524305)

[3.3.2 根据要求绘制目标框与轨迹 10](#_Toc197524306)

[第4章 讨论与改进 10](#_Toc197524307)

[4.1算法优势分析 10](#_Toc197524308)

[4.1.1 算法速度优势 10](#_Toc197524309)

[4.1.2 鲁棒性较高 10](#_Toc197524310)

[4.1.3 可视化效果清晰 11](#_Toc197524311)

[4.2 现存问题 11](#_Toc197524312)

[4.2.1 短时遮挡目标仍偏移 11](#_Toc197524313)

[4.2.2 小目标识别不稳定 11](#_Toc197524314)

[4.2.3 无法解决拍摄镜头发生角度突转的问题 12](#_Toc197524315)

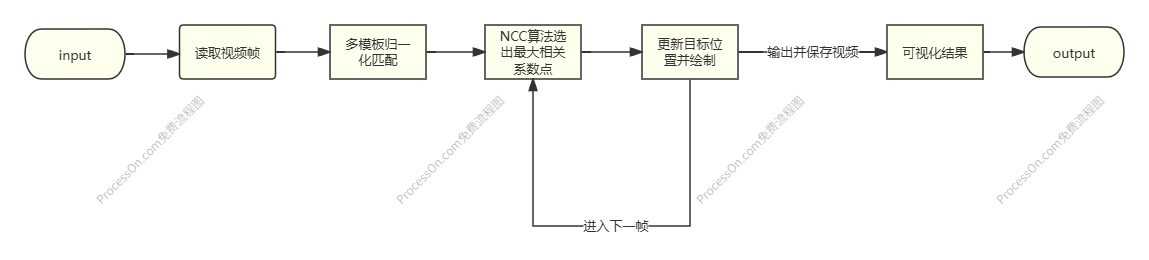
[4.2.4 目标检测抖动问题 12](#_Toc197524316)

[4.3未来改进方向 12](#_Toc197524317)

[结论 13](#_Toc197524318)

[附录 13](#_Toc197524320)

第1章 方法设计

1.1算法流程图

1.2跟踪策略

对于案例1，先进行全图匹配，通过NCC算法（归一化互相关）/SSD算法（差值平方和）计算相关值确定图像第一帧位置后，采用局部搜索的策略进行目标跟踪，在上一帧位置的周围20像素进行目标匹配，目标跟踪稳定性强。

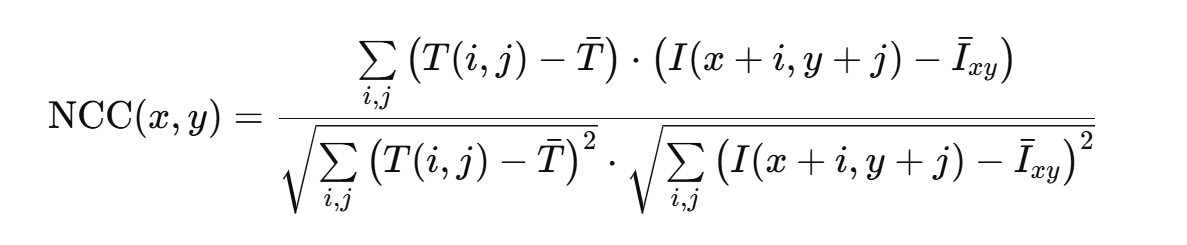
对于案例2，由于复杂背景影响，先在视频帧中提前提取多个模板用于目标匹配，手动提供初始位置，随后进行局部搜索，但当目标出现完全遮挡时无法匹配目标。因此引入卡尔曼滤波器进行运动预测，效果有微小提升，但当目标遮挡时间过长后。

对于案例3，连续运动的目标太小，且原视频和模板像素不高，即使提供初始位置，使用NCC/SSD算法也难以进行连续跟踪，极容易出现漏检、误检、无法识别的现象。因此采用卡尔曼滤波器预测目标运动的方向以提高小目标在低分辨率视频中的跟踪稳定性。

1.3核心算法说明

1.3.1 NCC（归一化互相关）

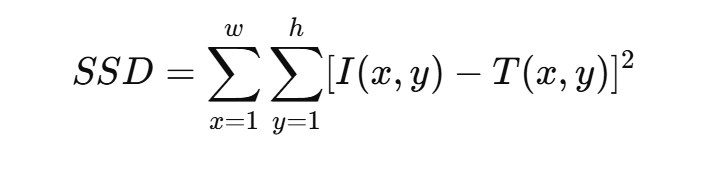
归一化互相关（Normalized Cross-Correlation, NCC）是一种用于模板匹配和图像相似度计算的经典方法。它解决了图像亮度和对比度变化对匹配准确性的影响，适合在灰度或光照变化不大的情况下使用。

 NCC算法的计算公式如下：

计算出模板和检测视频帧的协方差后进行归一化处理，计算出的互相关值最大的窗口就是模板的最大匹配位置，即目标位置。

1.3.2 SSD（差值平方和）

SSD（Sum of Squared Differences）是一种图像相似性度量方法，常用于模板匹配、立体匹配和图像配准等计算机视觉任务中。其基本思想是：通过逐像素比较两幅图像（或图像块）之间的差异，将每个像素的差值平方后求和，结果越小表示两者越相似。

SSD 算法的计算公式如下：

计算出模板和检测视频帧的差值平方和，计算值最小的窗口即为最佳匹配位置，即目标位置。

1.3.3 卡尔曼滤波

在本项目中，为了解决视频中目标存在短时遮挡或检测失败的问题，笔者采用了卡尔曼滤波器（Kalman Filter）。卡尔曼滤波是一种基于状态空间模型和贝叶斯估计理论的递推滤波算法，能够利用目标历史的位置信息（如速度、加速度等）预测其未来的运动轨迹，并对当前的检测结果进行优化和平滑。

在案例2目标出现短时遮挡、光照变化或姿态突变时，简单的NCC匹配方法会漏检或误检，此时卡尔曼滤波器能根据历史状态预测目标大致位置，从而避免匹配误检测其他车辆或背景干扰目标，实现鲁棒的连续跟踪。检测可信时，滤波器会用新的观测数据更新预测；检测不可信时，则保持预测状态，从而减弱检测失败带来的追踪抖动。

1.3.4 多模板匹配

在案例1与案例2中，目标会发生旋转、大小缩放等姿态突变的情况，或摄像视角会发生偏移、旋转等，导致模板与目标难以匹配，为此采用多模板匹配算法，增加目标匹配的鲁棒性与稳定性，提高目标检测与跟踪的准确率。

先用target.py文件手动标注视频帧内目标位置，再通过generate\_template.py文件在视频帧中截取模板并保存，最后在主文件中创造模板列表template\_list，进行循环匹配，得出归一化互相关值最高的最佳模板以及最佳匹配窗口位置，可视化结果并保存。

1.3.5 局部搜索

对于案例1，视频像素值较高，当对视频每帧进行全局搜索时代码运行速度较慢。因此，对算法进行改进，视频为第一帧时进行全局搜索定位目标，并存储目标位置，后续帧则在上一帧位置的周围20像素内进行局部搜索（考虑到目标位移/运动均为连续运动），最后运行结果良好。

对于案例2，目标的环境背景较复杂，干扰大，虽然运行速度较快，但若每帧进行全局搜索，容易错误定位目标，因此给出视频第一帧目标车辆的初始位置，后续视频帧则在第一帧基础上进行局部搜索，最后目标检测和跟踪的稳定性有很大提升。

1.3.6 模糊搜索

实验发现，在案例1中第一帧全局搜索仍然耗费较长的时间，对此对算法进行改进，将视频第一帧进行尺度缩小，匹配到目标的模糊位置后再将坐标进行还原，这样在保证目标检测精度的前提下提高了代码运行速度。

第2章 系统实现

2.1开发环境

CPU AMD Ryzen 7 5800HS Creator Edition

Visual Studio Code

Python 3.9.13

Numpy 1.24.4

Cv2 4.11.0

2.2代码结构与关键函数示例

*# 计算归一化互相关*

def **cross\_correlation**(image, template);

*# 计算差值平方和*

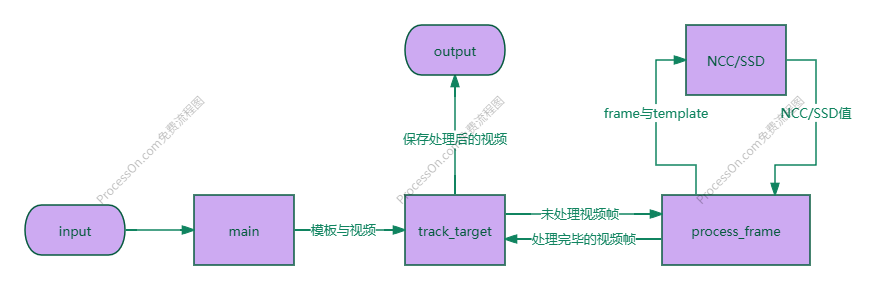
def **ssd\_match**(image\_patch, template);

*# 目标匹配与跟踪*

def **track\_target**(video\_path, template, output\_path="output.mp4");

*# 视频帧处理*

def **process\_frame**(frame, template\_list);

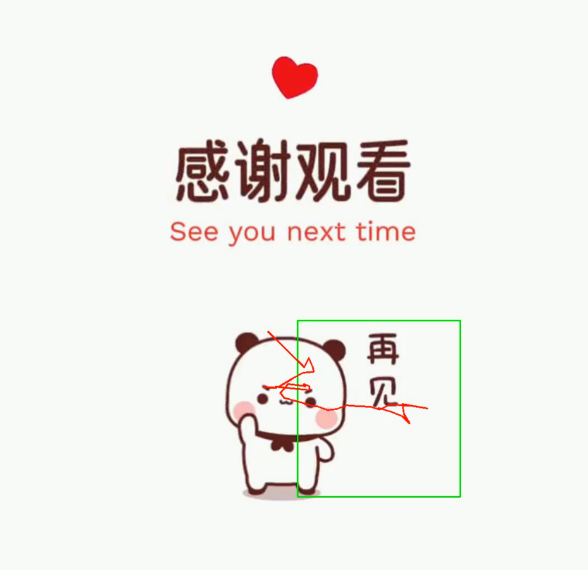
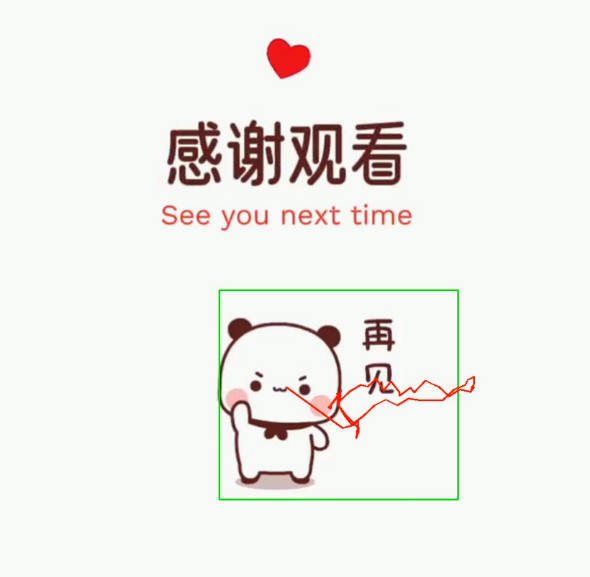
主程序中调用目标匹配与跟踪函数，在该函数中对视频进行处理，每帧调用process\_frame()，process\_frame()中调用计算NCC和SSD值的函数。

第3章 实验与结果

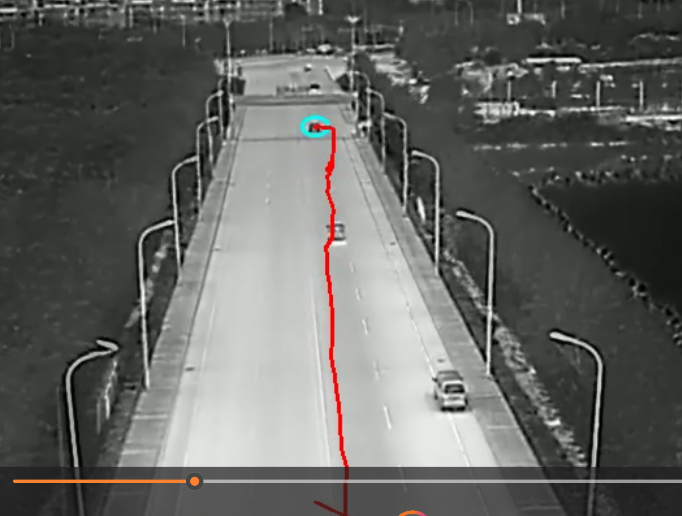
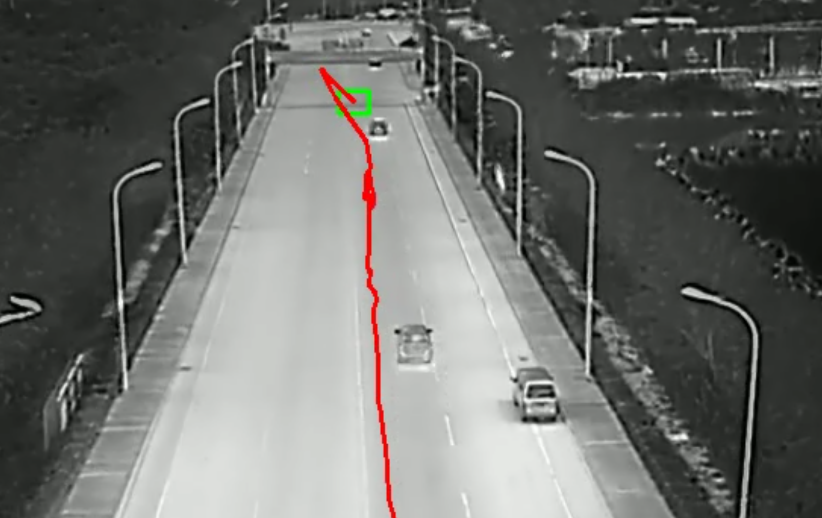
3.1测试案例对比

为验证本文方法在不同复杂场景下的鲁棒性，设置不同变量（单模板与多模板、NCC与SSD、是否加入卡尔曼滤波等）进行案例测试对比。

3.1.1 NCC与SSD

对案例1，采用两种算法进行目标匹配，左图为NCC算法，右图为SSD算法，目标在视频结尾出现间断的运动，SSD算法相较于NCC算法能更快速地匹配到目标位置，且可以看出SSD算法的目标运动轨迹抖动较小，比NCC算法鲁棒性更高。

3.1.2 不同模板选取

对于不同模板的选取，目标识别情况也有不同。提前提取如下6个模板，左图使用模板1、3、4，右图使用模板1、3、4、5，当目标车辆行驶到远处时，左图会发生目标丢失，而右图则可以稳定识别与跟踪。

3.1.3 是否使用卡尔曼滤波

案例2中，绘制出原始目标框（绿）、原始运动轨迹（黄）以及经过卡尔曼滤波器纠正后的目标运动（蓝），卡尔曼纠正后的目标运动轨迹（红）。在车辆经过桥梁的短时遮挡时，经过卡尔曼滤波器修正的目标识别相对更加稳定，但仍存在错检现象。

在案例3中，由于视频起始几帧摄像角度发生较大偏转，目标位移较大，是否加入卡尔曼滤波效果比案例2更加显著。图中黄色框为经过卡尔曼滤波器修正后的目标，可以看出黄框在镜头突转的情况下正确识别出骑车人，而未经过卡尔曼滤波器的原目标检测位置有较大偏差。

3.2量化性能分析

经过整合代码，对系统性能进行量化分析。对于定位精度，使用calculate\_stability()计算并检测帧间位置变化的标准差，对于漏检率，计算出归一化互相关的得分，设置阈值为0.45，若得分小于0.45，则被判断为漏检，将漏检帧数除以视频总帧数可以得出漏检率（由于项目未给出目标的标准位置，因此无法计算出误检率）。

|  |  |  |
| --- | --- | --- |
| 案例 | 帧间位置变化标准差 | 漏检率 |
| 视频1\_NCC | 14.45像素 | 8.27% |
| 视频1\_SSD | 18.18像素 | 3.22% |
| 视频2\_NCC | 8.41像素 | 30.46% |
| 视频2\_卡尔曼 | 6.90像素 | 26.87% |

*# 计算帧间位置变化的标准差*

def **calculate\_stability**(points):

    if **len**(points) < 2:

        return 0.0

    points = **np**.**array**(points)

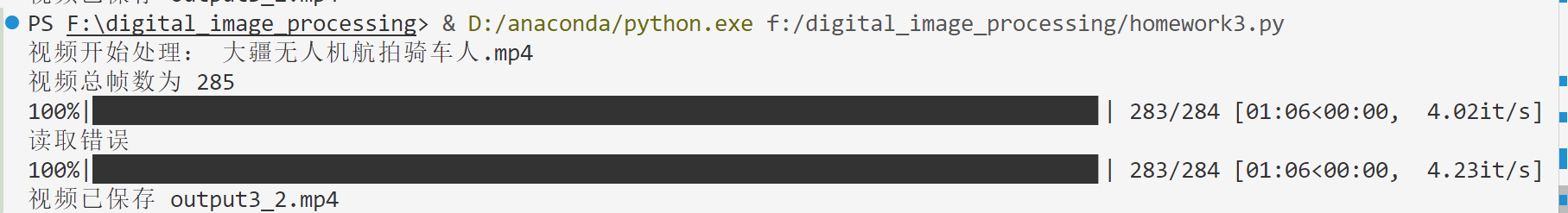
    diffs = **np**.**linalg**.**norm**(**np**.**diff**(points, axis=0), axis=1)

    return **np**.**std**(diffs)

通过观察帧间变化标准差指标，可以看出，视频1使用NCC和SSD两种算法的目标框抖动程度不同，SSD算法在视频结尾进行了较大平移，导致帧间变化标准差比NCC算法更大，视频2中，通过添加卡尔曼滤波器修正，帧间位置变化标准差有所下降，可见卡尔曼算法对目标检测的稳定性有提升。

通过观察漏检率，可以看出，视频1中SSD算法相较于NCC算法更稳定，这是因为在视频最后目标出现不连续运动，而SSD算法能够更快地定位到精准位置，视频2中卡尔曼算法对目标车辆的检测精度也有所提升。

3.3可视化效果

3.3.1 视频帧处理

在视频处理时，编写代码显示视频处理的进度条，可以看到视频总帧数、处理帧数、处理时间以及处理速度等指标。

对每个案例视频，将处理后的视频帧保存为本地MP4视频，将目标用规定颜色框标出，并绘制出目标的运动轨迹。

3.3.2 根据要求绘制目标框与轨迹

在3.1测试案例对比中已有详细展示，此处不再赘述。

第4章 讨论与改进

4.1算法优势分析

4.1.1 算法速度优势

通过局部搜索与模糊搜索，在未使用GPU加速的情况下极大提升了算法的运行速度，原先案例1视频处理时间高达几十分钟甚至一小时，优化后只需5-10分钟。

4.1.2 鲁棒性较高

多模板与卡尔曼滤波器极大提升了算法的稳定性与鲁棒性，如在案例2中，通过可视化结果，可以看出算法漏检率和误检率都有所下降。

4.1.3 可视化效果清晰

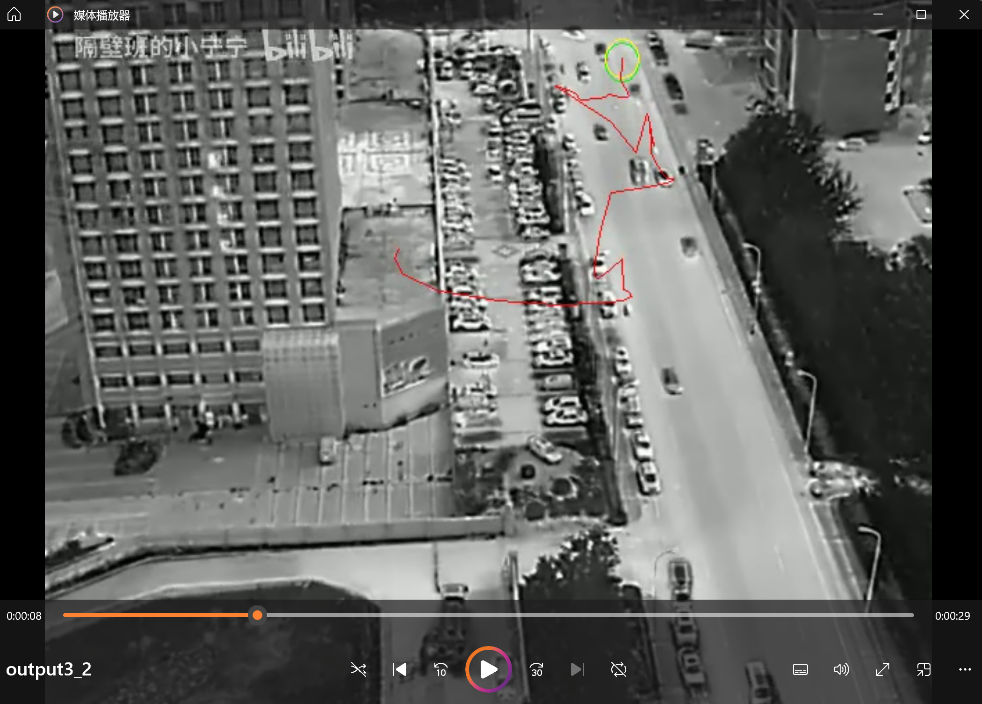
通过框选目标以及绘制目标运动轨迹，可以极其直观看出目标检测与跟踪算法的稳定性与鲁棒性。

4.2 现存问题

4.2.1 短时遮挡目标仍偏移

 在案例2中，当目标车辆经过桥梁时会出现短时完全遮挡的情况，即使使用卡尔曼滤波后对目标车辆的位移进行预测，但由于道路上相似车辆干扰较多，卡尔曼预测的纠正效果持续时间不长，仍然会出现误检现象，直到目标车辆不再遮挡后才能够正确识别目标。下图中紫色框部分则为目标被桥梁遮挡后的不稳定识别结果（轨迹）。

4.2.2 小目标识别不稳定

 在案例3中，由于视频帧及目标像素值过低且背景干扰因素过多，无法进行稳定识别，在实践过程中我尝试了多种方式，如提前设定初始值、对模板和视频帧进行直方图均衡化处理、加入卡尔曼滤波器、增加模板数量、局部搜索等，仍然无法进行稳定的目标跟踪。

4.2.3 无法解决拍摄镜头发生角度突转的问题

通过观察案例视频，在案例3视频的起始几帧，摄像无人机镜头发生较大的角度偏转，对识别目标运动的轨迹有很大干扰，为此加大局部搜索范围，但同时受到背景噪声干扰的概率提高，增加了小目标检测的不确定性，降低了算法的鲁棒性。

4.2.4 目标检测抖动问题

在案例1与案例2中，由于多模板的模板大小不一，在进行最佳模板匹配时目标检测框会出现抖动，导致目标运动轨迹不够平滑（目标运动轨迹由检测框的中心位置绘制而成），频繁出现曲线与折线。

4.3未来改进方向

对于现在仍存在的诸多问题，考虑如下改进：对于短时目标遮挡仍不稳定的问题，考虑改进卡尔曼滤波器，提高帧间预测稳定性，增强卡尔曼滤波的修正效果；对于小目标识别不稳定情况，可对视频帧及模板进行图像预处理，增强目标特征，过滤环境背景噪声，提高小目标识别的稳定性；对于摄像镜头发生角度突转的问题，考虑引入光流法，在每一帧之间估计摄像头的全局运动，进行全局运动补偿，或扩大搜索区域；对目标检测抖动问题，考虑改进匹配算法的精度，并使用自适应模板更新，减少模板匹配的误差。

结论

通过本项目实验，我学习到了如何在不使用AI算法的情况下进行目标检测与跟踪，让我更了解到该类算法的底层逻辑以及不同算法的优势与劣势，提升了我的代码编写与debug能力，收获颇丰，希望未来能够对自己的算法系统进行进一步的改进。

附录

Gitub网址：<https://github.com/YESHUOYUE/digital_image_processing>

百度网盘输出视频：通过网盘分享的文件：102204223陈亚欣数字影像处理输出视频

链接: https://pan.baidu.com/s/1iZrN3V72kj3-eog\_bU4sTQ 提取码: 5114